Animation Project in C++

Due Time: 23.59, Sat 14 October 2017 **Earnings:** 8% of your final grade

NOTE: Plan to finish a few days early to avoid last minute hardware/software holdups for which no allowance is given.
NOTE: The code in this assignment must be your own work. It must not be code taken from another student or written for you by someone else, even if you give a reference to the person you got it from (attribution); if it is not entirely your own work it will be treated as plagiarism and given a fail mark, or less.

Purpose: This is a continuation of assignment 0 but now written in the C++ language. It works the same. It is a console application that holds the data of an animation application (there is no actual animation graphics in the assignment) using a forward list (aka singly-linked list) of unspecified length in dynamic memory for its data.
Part of the code is shown on the next page; it is also on Blackboard in a text file that you can copy and paste. You **MUST** use this code **without modification (not a single character changed): no code added or removed, no new global variables or functions, no new classes, no macros, no defines and no statics**. Your task is to implement, using C++, only the Animation and Frame class member functions and not add any new ones. Everything you write is in the files Animation.cpp and Frame.cpp.

The Animation is a series of Frames held in a forward list. When the list runs it displays the details of each Frame at intervals of 1 second using the system clock. You can:

- Add a new Frame to the Animation at a position selected by the user
- Delete all the Frames in the Animation
- Run the Animation to show the list of Frame details one after another at 1 second intervals
- Quit

An example of the output of the running application is given at the end. Yours must work identically and produce identical output.

Note the following:

- dynamic memory management is done with new and delete (not malloc, realloc or free) and at any instant there is never any unused dynamic memory – only allocate what you need
- input and output is done with cin and cout
- you can only use functions like strlen() and strcpy() or similar etc. from the standard library to handle strings of null terminated chars (names can be up to 256 chars long)
- Release of dynamically allocated memory is done in destructors so there is no resource leak (or you lose 30%).

See the Marking Sheet for how you can lose marks, but you will lose at least 60% if:
1. you change the supplied code in any way at all (not a single character) - no code added or removed, no macros, no defines, no statics and no additional classes, global functions or variables,
2. it fails to build in Visual Studio 2015,
3. It crashes in normal operation (such as running an empty list etc.),
4. it doesn't produce the example output.

Part of the code is shown on the next page. You MUST use this code **without modification.** Your task is to add the implementation of the class member functions.

What to Submit : Use Blackboard to submit this assignment as a zip file (**not** RAR) containing only Animation.cpp and Frame.cpp. The name of the zipped folder **must** contain your name as a prefix so that I can identify it, for example using my name the file would be tyleraAss1CST8219.zip. It is also vital that you include the Cover Information (as specified in the Submission Standard) as a file header in your source file so the file can be identified as yours. Use comment lines in the file to include the header.

Before you submit the code,

- check that it builds and executes in Visual Studio 2015 as you expect - if it doesn't build for me, for whatever reason, you get a deduction of at least 60%.
- make sure you have submitted the correct file – if I cannot build it because the file is wrong or missing from the zip, even if it's an honest mistake, you get 0.

There is a late penalty of 25% per day. Don't send me files as an email attachment – they will get 0.

Supplied code (also in a text file you can copy and paste on BlackBoard). Don't change it.

```
//Animation.h
#pragma once

class Animation {
          Frame* frames;
public:
          Animation();
          ~Animation();
          void InsertFrame();
          void DeleteFrames();
          void RunFrames();
};
```

```
// Frame.h
#pragma once

class Frame {
          char* fileName;
          Frame* pNext;
public:
          Frame();
          ~Frame();
          char*& GetfileName() { return fileName; }
          Frame*& GetpNext(){ return pNext; };
};
```

```
// ass1.cpp
#define _CRT_SECURE_NO_WARNINGS
#define _CRTDBG_MAP_ALLOC     // need this to get the line identification
//_CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF|_CRTDBG_LEAK_CHECK_DF); // in main, after local declarations
//NB must be in debug build

#include <crtdbg.h>
#include "Frame.h"
#include "Animation.h"
#include <iostream>
using namespace std;

bool running = true;

int main(void)
{
          char selection;
          bool running = true;
          Animation A;
          _CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF|_CRTDBG_LEAK_CHECK_DF); // in main, after local declarations

          while (running)
          {
                    cout<<    "MENU\n 1. Insert a Frame\n 2. Delete all the Frames\n 3. Run the Animation\n 4. Quit\n"<<endl;
                    cin>>selection;

                    switch (selection)
                    {
                    case '1':
                              A.InsertFrame();
                              break;
                    case '2':
                              A.DeleteFrames();
                              break;
                    case '3':
                              A.RunFrames();
                              break;
                    case '4':
                              running = false;
                              break;
                    default:
                              break;
                    }
          }

          return 0;
}
```

Example Output:

```
MENU
 1. Insert a Frame
 2. Delete all the Frames
 3. Run the Animation
 4. Quit

1
Insert a Frame in the Animation

Please enter the Frame filename: Animation_1
This is the first Frame in the list
MENU
 1. Insert a Frame
 2. Delete all the Frames
 3. Run the Animation
 4. Quit

1
Insert a Frame in the Animation

Please enter the Frame filename: Animation_2
There are 1 Frame(s) in the list. Please specify the position (<= 1 to insert at: 1
MENU
 1. Insert a Frame
 2. Delete all the Frames
 3. Run the Animation
 4. Quit

1
Insert a Frame in the Animation

Please enter the Frame filename: Animation_3
There are 2 Frame(s) in the list. Please specify the position (<= 2 to insert at: 0
MENU
 1. Insert a Frame
 2. Delete all the Frames
 3. Run the Animation
 4. Quit

3
Run the Animation
Frame #0, time = 1 sec
Image file name = Animation_3
Frame #1, time = 2 sec
Image file name = Animation_1
Frame #2, time = 3 sec
Image file name = Animation_2
MENU
 1. Insert a Frame
 2. Delete all the Frames
 3. Run the Animation
 4. Quit

2
Delete all the Frames from the Animation
MENU
 1. Insert a Frame
 2. Delete all the Frames
 3. Run the Animation
 4. Quit

3
No frames in the animation
MENU
 1. Insert a Frame
 2. Delete all the Frames
 3. Run the Animation
 4. Quit
```